“¿Quieres ser tu propio jefe?”

Laura Ortiz Merchán

Daniel Morales Ramírez

Kevin Canchila Rodríguez

Juan José Reina Reyes

Javier Casas Salgado

Universidad del Rosario

Ingeniería de Datos

2024-S1

27 de abril de 2024

Problema

Con base en los datos seleccionados sobre inversiones en fondos de inversión en Colombia durante el período del 1 al 31 de marzo, se pueden definir varios problemas de datos que pueden ser respondidos:

1. Identificación de tendencias de inversión: Se pueden identificar las tendencias durante el mes de marzo, como los sectores o tipos de fondos que experimentaron un mayor interés por parte de los inversores. Esto puede ayudar a prever futuros movimientos del mercado y ajustar estrategias de inversión en consecuencia.
2. Análisis de la distribución de inversiones por entidad: Se puede analizar cómo se distribuyen las inversiones entre diferentes entidades financieras, identificando las entidades que recibieron la mayor cantidad de inversión y comparando su desempeño con el de otras entidades.
3. Gestión de riesgos: Se puede realizar un análisis de riesgos para evaluar la exposición de los fondos de inversión a diferentes factores de riesgo, como el riesgo de mercado, el riesgo crediticio y el riesgo de liquidez. Esto puede ayudar a los inversores a tomar decisiones informadas sobre la diversificación de sus carteras y la gestión de riesgos.
4. Segmentación de inversores: Se pueden identificar diferentes segmentos de inversores según su comportamiento de inversión, como inversores conservadores, moderados y agresivos. Esto puede ayudar a las entidades financieras a personalizar sus estrategias de marketing y ofrecer productos de inversión adaptados a las necesidades de cada segmento.

Reglas

1. Las Entidades son únicas, por tanto están identificadas como tal.
2. Una Entidad puede poseer inversiones en distintos negocios.
3. Los Negocios son Fondos de Inversiones, caracterizados por el tipo de Fondo que son.
4. Los Negocios son únicos, por tanto están identificados con un código.
5. Los Negocios pueden ser su rama principal o un compartimiento del negocio.
6. Cada Negocio genera una rentabilidad, la cual está tipificada por un periodo de tiempo.
7. Cada Negocio tiene un número de inversiones hechas.

Descripción

Entidad: Entidad

- Atributos:

- id_Movimientos (int) (FOREIGN KEY relacionada con Identificación)
- Fecha (date)
- nombre_entidad (varchar)
- cod_negocio (int) (FOREIGN KEY relacionada con Negocio)
- cod_transaccion (int) (FOREIGN KEY relacionada con Rentabilidad)

- Relaciones:

- Relacionada a través de Negocio y Rentabilidad

Entidad: Identificación

-Atributos:

- tipo_entidad (int)
- cod_entidad (int)
- Nombre_entidad (varchar) (PRIMARY KEY)

- Relaciones:

-Relacionada con Entidad

Entidad: Negocio

- Atributos:

- cod_negocio (int) (PRIMARY KEY)
- Nombre_negocio (varchar)
- sub_negocio (int) (FOREIGN KEY relacionada con SubTipo)
- idPC (int) (FOREIGN KEY relacionada con Jerarquía)

- Relaciones:

- Relacionada con SubTipo y Jerarquía, en este caso Negocio es la Entidad Fuerte.

Entidad: Subtipo

- Atributos:

- id_ST (int) (PRIMARY KEY)
- Nombre_subtipo (varchar)

- Relaciones:

-Relacionada con Negocio

Entidad: Jerarquía

-Atributos:

- id_PC (int) (PRIMARY KEY)
- PC (varchar)

-Relaciones:

-Relacionada con Negocio

Entidad: Rentabilidad

- Atributos:

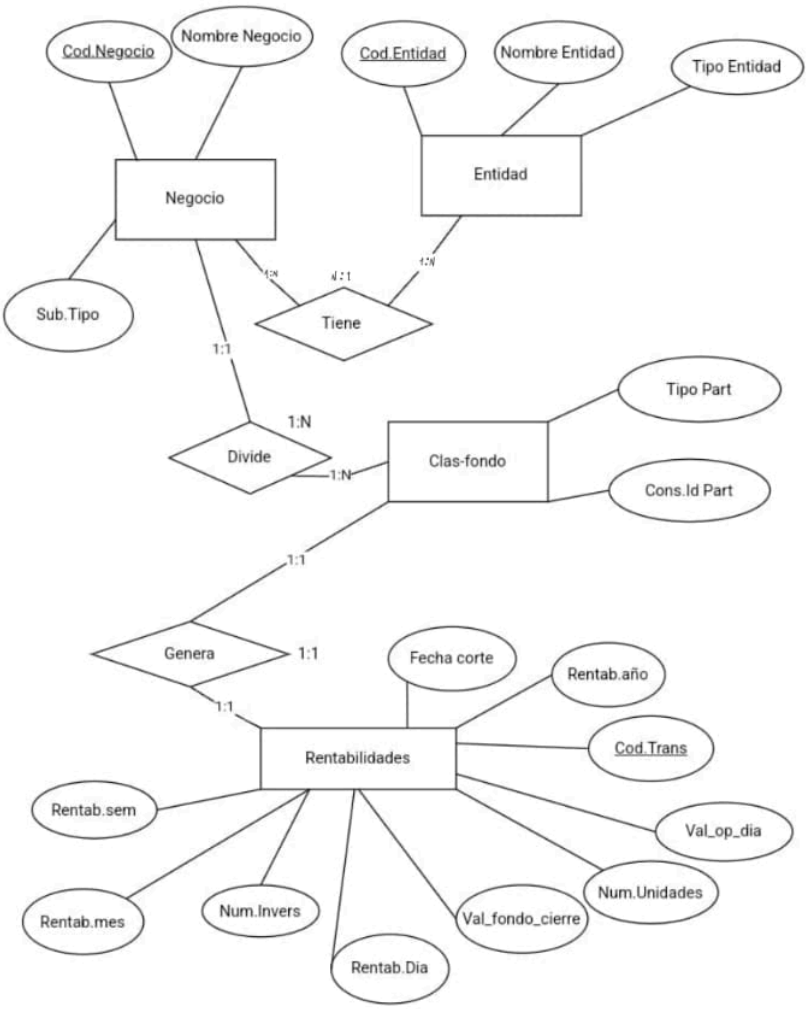
- cod_transaccion (int) (PRIMARY KEY)
- num_unidades (int)
- valor_unidad (float)
- valor_fondo_cierre (float)
- num_invers (int)
- rentab_dia (float)
- rentab_mes (float)
- rentab_sem (float)
- rentab_mes (float)
- rentab_año (float)

- Relaciones:

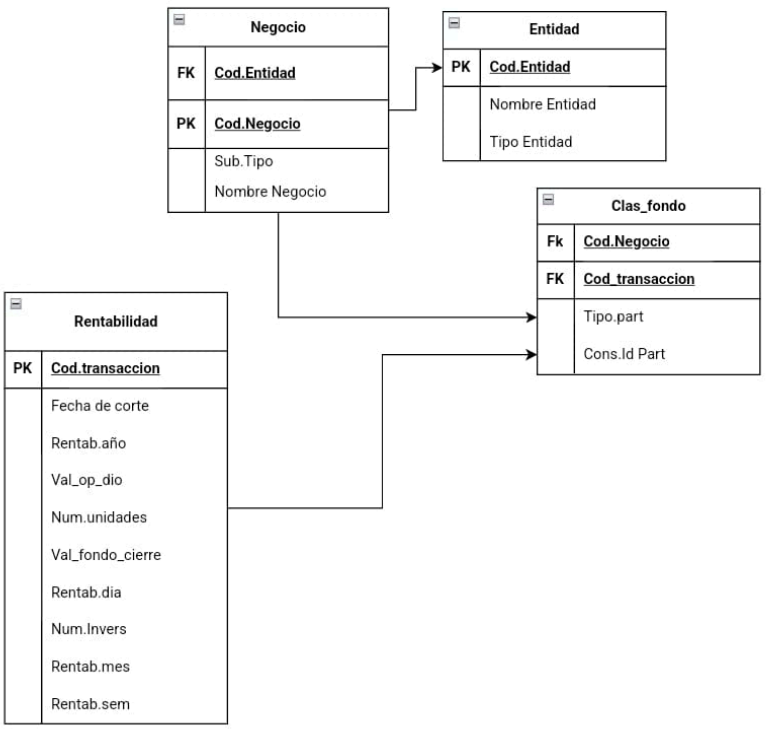
- Relacionada con Entidad

Estas entidades, atributos y relaciones proporcionan una estructura básica para entender la organización de los datos relacionados con las inversiones en fondos de inversión en Colombia.

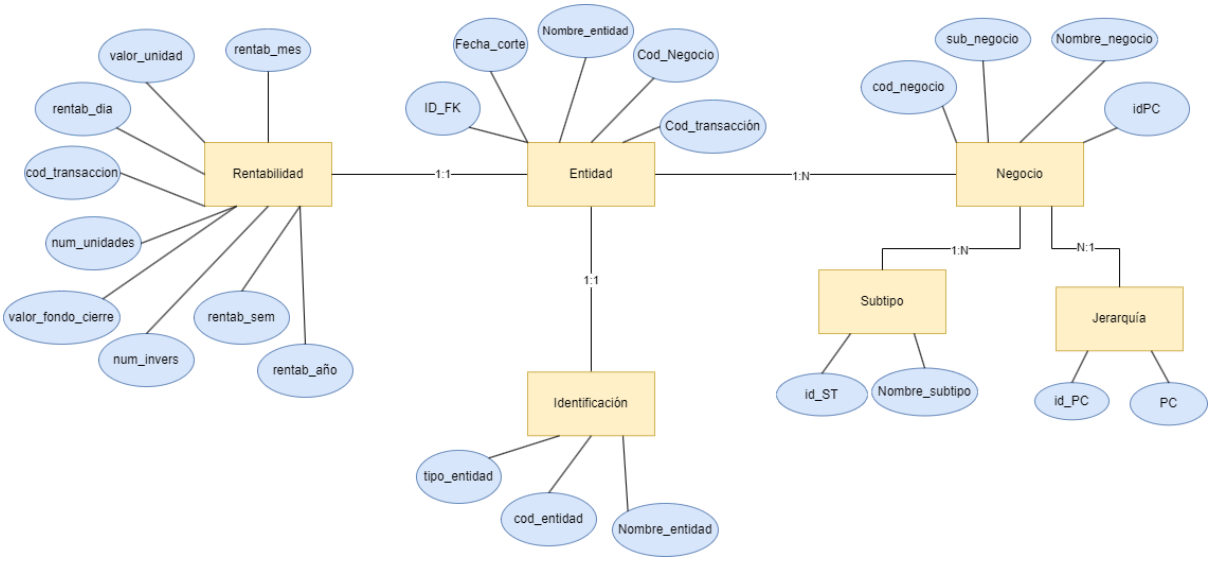
Modelo ER



Modelo relacional



Modelo Relacional normalizado:



Descripción carga de información en la base de datos:

Para la carga masiva de datos establecimos la siguiente base de datos en PostgreSQL mediante el query:

create table Entidad (
id_Movimientos int primary key,
Fecha date,
Nombre_entidad varchar(100),
cod_negocio int,
cod_transaccion int ,
foreign key (Nombre_entidad) references identificacion(Nombre_entidad),
foreign key (cod_transaccion) references Rentabilidad(cod_transaccion),
foreign key (cod_negocio) references Negocio(cod_negocio)
);

create table Identificacion(
tipo_entidad int,
cod_entidad int,
Nombre_entidad varchar(100) primary key
);

create table Negocio(
cod_negocio int primary key,
Nombre_negocio varchar(100),

```
sub_negocio int,
idPC int,
foreign key (sub_negocio) references Subtipo(id_ST),
foreign key (idPC) references Jerarquia(id_PC)
);


create table Subtipo(
id_ST int primary key,
Nombre_subtipo varchar(100)
);


create table Jerarquia(
id_PC int primary key,
PC varchar(100)
);


create table Rentabilidad(
cod_transaccion int primary key,
num_unidades float,
valor_unidad float,
valor_fondo_cierre float,
num_invers int,
rentab_dia float,
rentab_mes float,
rentab_sem float,
rentab_año float
);
```

Por otro lado, la base de datos original que obtuvimos, un archivo .xls (archivo de excel), a la hora de normalizar la base de datos hasta su tercera forma normal, separamos cada entidad en un documento distinto. Por tanto, como resultado obtuvimos 6 archivos .csv (valores separados por comas) cada uno de la entidad correspondiente. Al ser estos nuestros archivos de carga la sentencia SQL resultó:

(para la carga hay que establecer la ruta de la carpeta ‘Final’ del repositorio)

```
copy public.Subtipo from 'C:\Users\promaster13031\Desktop\DATOS\Subtipo.csv' WITH
(FORMAT csv, DELIMITER ';', HEADER);
```

copy public.Rentabilidad from 'C:\Users\promaster13031\Desktop\DATOS\Rentabilidad.csv' WITH (FORMAT csv, DELIMITER ';', HEADER);

copy public.Jerarquia from 'C:\Users\promaster13031\Desktop\DATOS\Jerarquia.csv' WITH (FORMAT csv, DELIMITER ';', HEADER);

copy public.Negocio from 'C:\Users\promaster13031\Desktop\DATOS\Negocio.csv' WITH (FORMAT csv, DELIMITER ';', HEADER);

COPY public.Identificacion FROM 'C:\Users\promaster13031\Desktop\DATOS\Identificacion.csv' WITH (FORMAT csv, DELIMITER ';', HEADER);

copy public.Entidad from 'C:\Users\promaster13031\Desktop\DATOS\Entidad.csv' WITH (FORMAT csv, DELIMITER ';', HEADER);

Posibles Escenarios de Análisis:

1. **Análisis de Rendimiento Histórico:** Examinar el rendimiento histórico de cada negocio y entidad para asegurar decisiones basadas en datos probados.
2. **Diversificación de Riesgos:** Mantener la diversificación en las inversiones del fondo, asegurando que ninguna entidad o negocio representa una parte desproporcionada del portafolio total.
3. **Criterios de Selección:** Definir criterios claros y objetivos para la selección de negocios y entidades, incluyendo rentabilidad esperada, estabilidad financiera, y potencial de crecimiento.
4. **Análisis de Liquidez y Solvencia:** Evaluar la capacidad de las entidades y negocios para cumplir con sus obligaciones financieras a corto y largo plazo. Esto implica analizar ratios financieros como el ratio de liquidez, el ratio de endeudamiento, el ratio de cobertura de intereses y el ratio de solvencia para determinar la salud financiera y la capacidad de generar flujos de efectivo suficientes.

Desarrollo de conexión de módulos a la base de datos con python:

```
import psycopg2 # Importamos la librería psycopg2

try:
    # Intentamos establecer una conexión con la base de datos
PostgreSQL
    connection = psycopg2.connect(
        host='localhost',  # Dirección del servidor de base de datos
        user='postgres',  # Nombre de usuario
```

```
        password='123456789',  # Contraseña del usuario
        database='Reporte_financiero_2024_03',  # Nombre de la base de datos a la que queremos conectarnos
        port='5433', # Puerto en el que está escuchando el servidor de base de datos
    )
    print("Conexión exitosa") # Si la conexión es exitosa, imprimimos un mensaje
    cursor = connection.cursor() # Creamos un objeto cursor para ejecutar comandos SQL


    # Tabla1 Entidad
    cursor.execute("SELECT * FROM Entidad")
    rows_tabla_Entidad = cursor.fetchall()
    for row in rows_tabla_Entidad:
        print(row)

    # Tabla2 Negocio
    cursor.execute("SELECT * FROM Negocio")
    rows_Negocio = cursor.fetchall()
    for row in rows_Negocio:
        print(row)

    # Tabla3 Identificacion
    cursor.execute("SELECT * FROM Identificacion")
    rows_Identificacion = cursor.fetchall()
    for row in rows_Identificacion:
        print(row)

    # Tabla4 Subtipo
    cursor.execute("SELECT * FROM Subtipo")
    rows_Subtipo = cursor.fetchall()
    for row in rows_Subtipo:
        print(row)

    # Tabla5 Jerarquia
    cursor.execute("SELECT * FROM Jerarquia")
    rows_Jerarquia = cursor.fetchall()
    for row in rows_Jerarquia:
        print(row)

    # Tabla6 Rentabilidad
```

```
    cursor.execute("SELECT * FROM Rentabilidad")
    rows_Rentabilidad = cursor.fetchall()
    for row in rows_Rentabilidad:
        print(row)


except Exception as ex:
    print(ex) # Si ocurre algún error durante la ejecución del bloque try, lo capturamos y lo imprimimos

finally:
    connection.close() # Finalmente, independientemente de si hubo éxito o error, cerramos la conexión a la base de datos
    print("Conexión finalizada") # Imprimimos un mensaje para indicar que la conexión ha sido cerrada
```

Para ver el código, es el archivo llamado “conexion.py” en el repositorio.

Repositorio de GitHub:

El siguiente repositorio es el cual usamos para la implementación de nuestra base de datos y manejo de la información: https://github.com/QueenR1014/Ingenieria_Datos